NLP At Scale: вся правда о предобученных моделях

Дмитрий Меркушов

Antispam ML mail.ru group







1

Зоопарк правил

2

Зоопарк стораджей

3

Банк фич

4

Зоопарк ML

Антиспам в цифрах

20M

Happy DAU

- 1.5B

  писем в сутки

- 20+

  ML-систем

- 80%

  спама



Ключевые проблемы

- Быстрая адаптация спамеров
- Решения деградируют (сами и с помощью)
- Тренд: target на тексты





Почему именно NLP

- Новые тексты легко генерируются спамерами
- Понятный измеримый трейд-офф «синонимичность – стоимость»
- Все еще самый понятный способ messaging к аудитории





ML как adversary-фронтир

- Синергия паттернов
- Эволюция моделей
- Фокус ML-сообщества на тексты 🤗





Эволюция NLP в сервисе

1

Эволюция задач

2

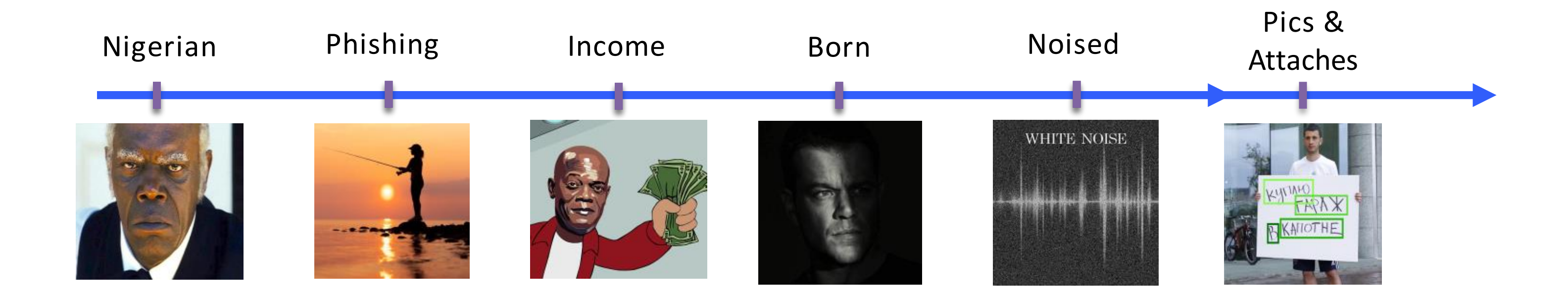
Эволюция подходов

3

Ограничения

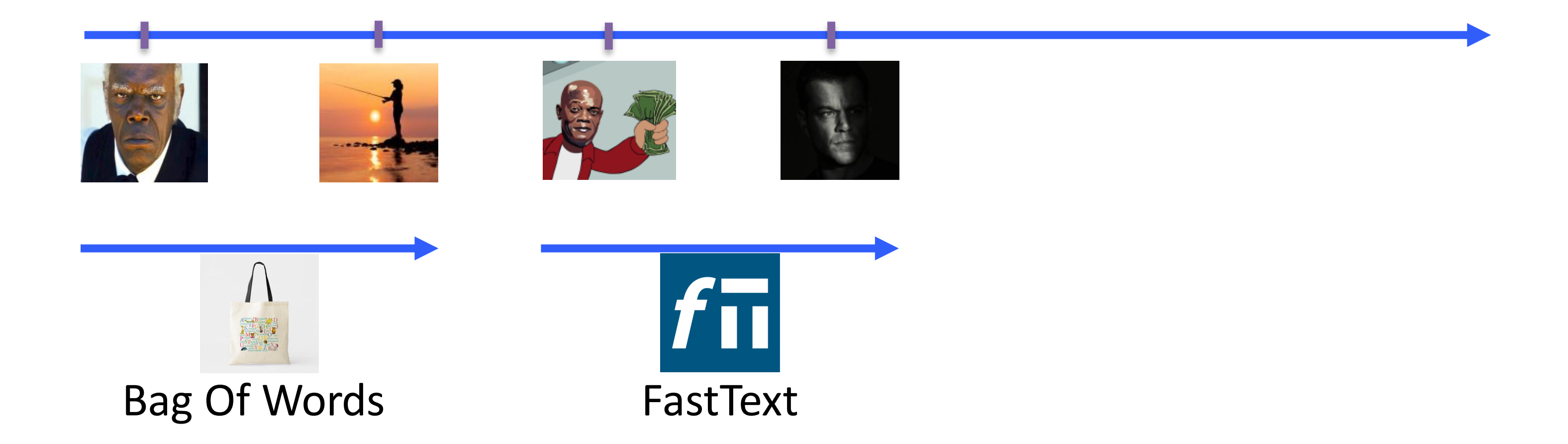


Таймлайн


Nigerian
Phishing
Income
Born
Noised
Pics &
Attaches
WHITE NOISE


HL
HighLoad++
Весна 2021

ML-Таймлайн





ML-Таймлайн

BCE

CE

Bag Of Words

FastText

PyTorch

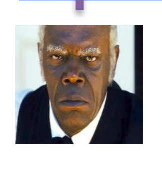


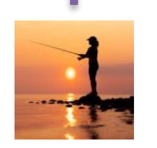
ML-Таймлайн

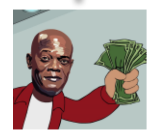
Before Chintala Era

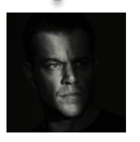


Chintala Era











Bag Of Words



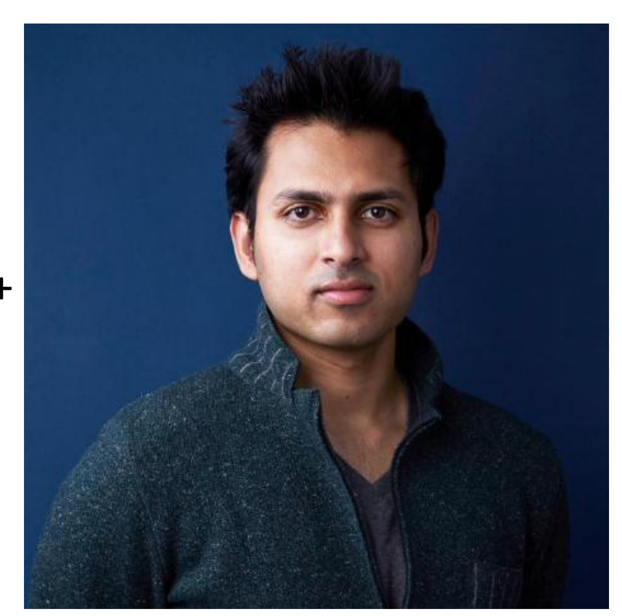
FastText





PyTorch

- Самый удобный фреймворк для глубокого обучения
- Быстрый inference и легкая интеграция с production C++
- Единый API для разных ML-архитектур





ML-Таймлайн

Deep Feed Forward (DFF)

Feed Forward

1D Convolution

Text CNN



Shallow Learning

1. Критерии
2. Сложности
3. Модели



Составляющие успеха









Новые слова

Адаптация под эволюцию словаря языка сервиса

Интерпретация

Насколько мы понимаем решение наших моделей

Учет контекста

Словарные конструкции меняют смысл в контексте

Attention важного

Некоторые части текста несут большую смысловую нагрузку



Bag of Words

- Архитектура - Линейный классификатор
  - Обучение – на supervised-метки
- Признаки – мера наличия слов (TF-IDF)
  - Количество - по размеру словаря сервиса
- Результат - взвешенная линейная комбинация





Bag of Words





Адаптация под эволюцию словаря языка сервиса



Интерпретация

Насколько мы понимаем решение наших моделей



Учет контекста

Словарные конструкции меняют смысл в контексте



Attention важного

Некоторые части текста несут большую смысловую нагрузку



FastText

- Архитектура – полносвязная неглубокая сеть
  - Обучение – на supervised-метки / на unsupervised-контекст
- Признаки – мера наличия слов / N-gram в окне текущего
- Результат - взвешенная линейная комбинация





FastText





Новые слова

Адаптация под эволюцию словаря языка сервиса



Интерпретация

Насколько мы понимаем решение наших моделей



Учет контекста

Словарные конструкции меняют смысл в контексте



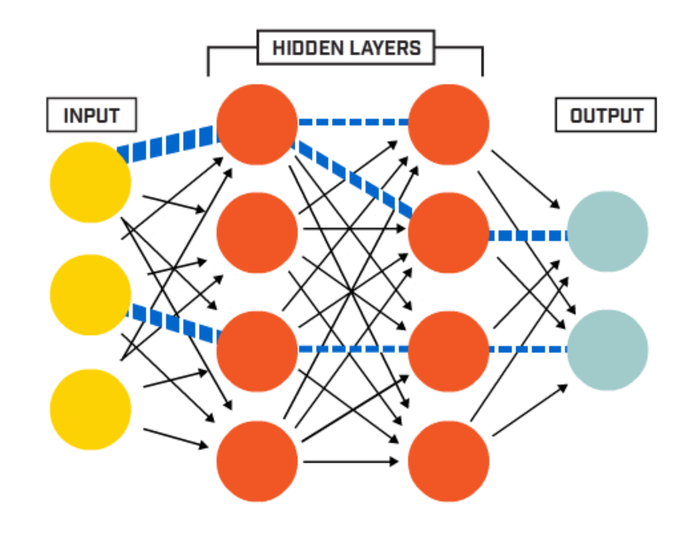
Attention важного

Некоторые части текста несут большую смысловую нагрузку



Feed Forward

- Архитектура – неглубокая сеть с нелинейностью
  - Обучение – на supervised-метки
- Признаки – усредненный FT-эмбеддинг слов
- Результат – softmax с последнего слоя





Feed Forward









Новые слова

Адаптация под эволюцию словаря языка сервиса

Интерпретация

Насколько мы понимаем решение наших моделей

Учет контекста

Словарные конструкции меняют смысл в контексте

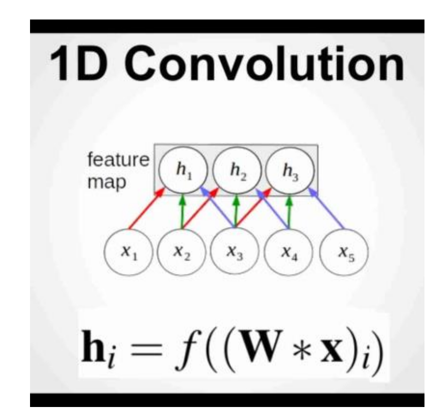
Attention важного

Некоторые части текста несут большую смысловую нагрузку



Сверточные сети

- Архитектура – глубокая (65k нейронов) сверточная сеть с механизмом 1d Convolution
  - Обучение – на supervised-метки
- Признаки – FT-эмбеддинги каждого слова
- Результат – AvgPooling по stride’у





Сверточные сети. Attention

Добрый день, Валентин Иванович! Меня зовут Вера, менеджер Вашего Банка. Вам предоставлена компенсационная выплата в размере 124560.34 руб

```
Most loyal
Банка. -0.011007905
Иванович! -0.009653926
день, -0.0067341924
Добрый -0.0057362914
Валентин -0.005717337


Most fraudulent
Вера, 0.0015135407
размере 0.003304243
Вам 0.011296153
выплата 0.017832875
компенсационная 0.030203879

Total score:
0.9788287
```





Сверточные сети



Новые слова

Адаптация под эволюцию словаря языка сервиса



Интерпретация

Насколько мы понимаем решение наших моделей



Учет контекста

Словарные конструкции меняют смысл в контексте



Attention важного

Некоторые части текста несут большую смысловую нагрузку



ML-Таймлайн

???

WHITE NOISE

Deep Feed Forward (DFF)

1D Convolution

$\mathbf{h}_i = f((\mathbf{W} * \mathbf{x})_i)$



ML-Таймлайн

???

WHITE NOISE

КУПЛЮ ГАРАЖ В КАПОТНЕ

Deep Feed Forward (DFF)

1D Convolution

feature map

$\mathbf{h}_i = f((\mathbf{W} * \mathbf{x})_i)$



Transformers. Why bother ?









Новые слова

Адаптация под эволюцию словаря языка сервиса



Интерпретация

Насколько мы понимаем решение наших моделей



Учет контекста

Словарные конструкции меняют смысл в контексте

Attention важного

Некоторые части текста несут большую смысловую нагрузку

Обобщаемость

Одна модель может решать разные задачи



Составляющие успеха

- Каждая отдельная модель – дорогая в обучении и поддержке
- Цель – дешевые Intent-классификаторы
- Решение – всю сложность инкапсулировать в признаках
- Признаки – одна **уберсложная** модель



Обобщаемость



Составляющие успеха

- Каждая отдельная модель – дорогая в обучении и поддержке
- Цель – дешевые Intent-классификаторы
- Решение – всю сложность инкапсулировать в признаках
- Признаки – одна **уберсложная** модель Transformer





Deep Dive. Свой Transformer

1

Семейство моделей

2

Зачем свой

3

Выборка и обучение

4

Продакшн



Introducing Transformers



Introducing Transformers



Introducing Transformers: RoBERTa

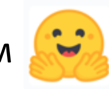
- Архитектура – очень глубокая (10-100м+) сеть с квадратичным механизмом self-attention
  - Обучение – на unsupervised-задачу masked-language-model
- Признаки
  - Byte-pair encoding токенизация
  - Positional encoding
- Результат – эмбеддинг специального CLS-токена



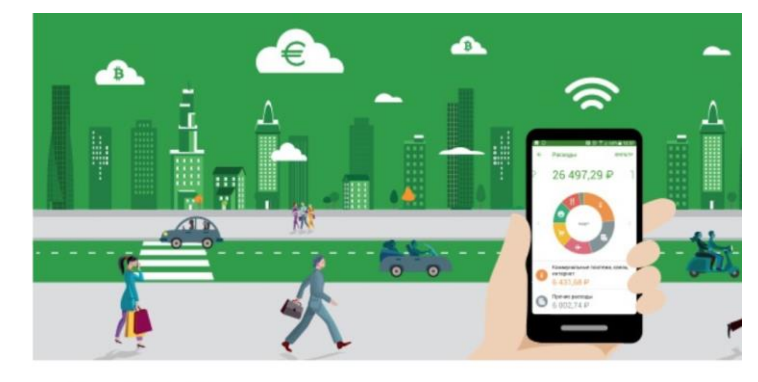


Предобученные модели

В открытом доступе есть модели на русском

- Подходят для большинства NLP-задач
- Качественно обучены сообществом 
    - Решена проблема ресурсов для обучения
- Некоторые реализации допускают дообучение модели на своих данных

Причем не только трансформеры, но и fastText







Однако

Специфика почты (и спама!) слишком специфична

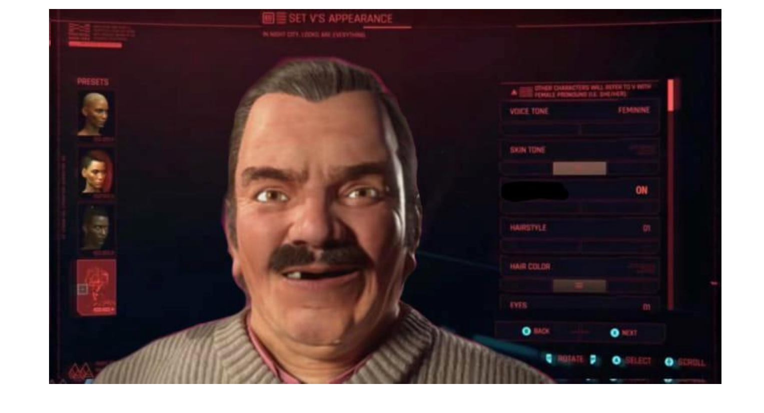
- Специфичный поток текстов – сложно назвать русским языком
- Специфичное внутреннее представление
- Как результат – готовые сложные модели доставляют качество на уровне собственных моделей меньшей сложности





Кастомизация

- Адресуем нужные продукту проблемы
    - Выборка из сервиса и выбор функции потерь
- Трейд-офф качества – производительности
    - Архитектура и параметры модели (i.e. attention heads, transformer layers, embedding size)
- Безопасность – у спамеров нет доступа к той же модели

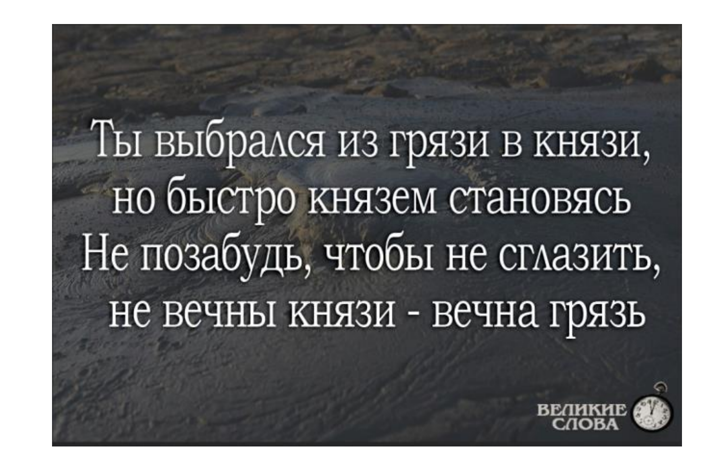




Transformer: from Zero to Hero

Челлендж из челленджей

- Инфраструктура для сбора выборки – большие модели needs more data (10M+ текстов)
- Инфраструктура для обучения – большие модели needs more GPU (8+ (16+, 32+, …) GPU), Network (20+ Гб канал)
- Инфраструктура для инференса – нужна вне зависимости от свой/чужой





Обучающая выборка

Помним главное - обобщаемость Unsupervised-модели на все типы спама

- Occam’s Razor – собрать как можно больше разнообразных данных с потока
    - Семплирование с потока по распределению вероятностей
    - Безостановочно - много дубликатов ограничивают скорость сбора
- Анонимизация такого корпуса текстов становится mandatory-фичей безопасности

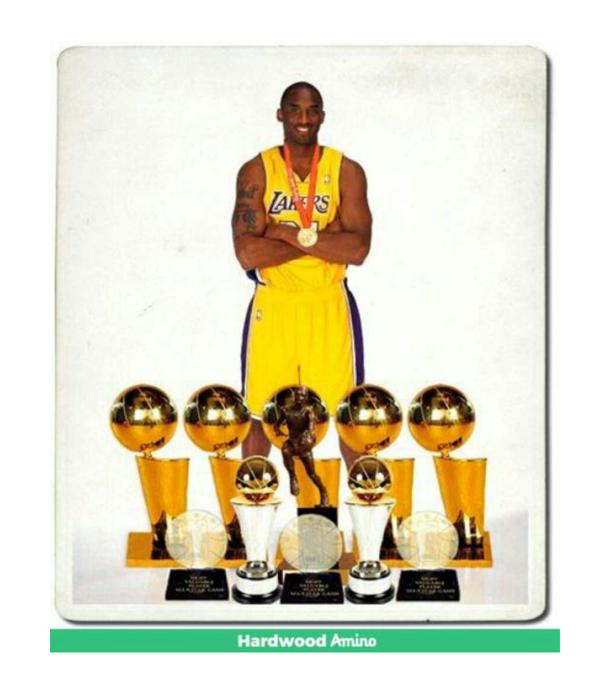




ML Research: MVPs

Сужаем воронку подходов локальными MVP

- Прорежаем список SOTA-подходов сообщества на соответствие целям задачи
- Обучение нескольких эпох на меньшем объеме данных
- Python dev-стенд для оценки инференса архитектуры
    - Python вызывает тот же libtorch.predict, что и плюсовый production-сервис
    - Я.Танк + TorchServe для имитации нагрузки

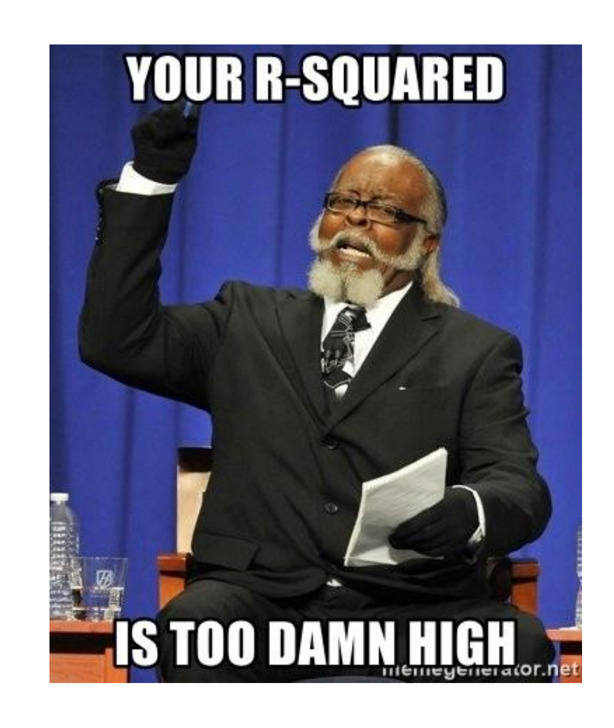




ML Research: критерии выбора

Важно выбрать пул локальных критериев

- Качественные метрики
  - Extrinsic- и intrinsic-оценки
  - На репрезентативных extrinsic-задачах (например, текущих intent-выборках)
- Технические метрики
  - Размер модели
  - Время inference, pre/post processing





Обучение

- Простой подход – выбрать машину с побольше GPU (например 8 x 2080 Ti, или A100 :)
- Но ограничены 1 машиной – немасштабируемая история
- Неизбежно построение GPU-кластера – отдельная большая логическая и техническая задача





Deployment

- Отдельный инференс–сервис в K8S, 1 под = 1 gpu
    - ответ нужен в онлайне = минимальный бюджет на latency
- Оптимизация алгоритмов токенизации (i.e. SentencePiece)
    - базовая логика O(N^2) разгоняется до O($NlogN$), если подумать
- SLA 99.99 оказался достижимой мечтой
    - Очередь + таймаут на вставку + ретрай = утилизация GPU до 80%





Как этим пользоваться

- Трансформер генерирует универсальный глубокий эмбеддинг письма
- Сверху обучается микроголова под нужный intent
  - Несколько слоев (FFN) или даже один (линейная)
  - Supervised – на размеченной выборке сильно меньшего объема





ML-эксплуатация

- Трансформер – монолит, но вносить изменения нужно редко (retrain с нуля или finetune)
- Intent головы – эксплуатировать привычно
  - Дообучение (finetune) с регуляризацией типа Knowledge Distillation
  - Новые паттерны через Feedback Loop
  - Деплой через A/B-тесты





CPU vs GPU

- GPU-Сервис
  - Трансформер
  - 1 inference на письмо
- CPU-Сервис
  - Intent-головы
  - Multiple inference на письмо
    - По количеству intent-моделей
    - Которые еще и A/B-тестируются





Сетап в цифрах. Scale

1Tb

Выборка обучения

- 100 GPU Tesla T4

Инференс трансформера

- 800 CPU cores

Инференс intent голов

- 24 GPU

Кластер обучения



Сетап в цифрах. Time

10ms

Медианный инференс трансформера

- < 1ms

Инференс головы

- 1 Неделя

Обучение модели с нуля

- 2 Месяца

Первичный сбор выборки

- 1 Квартал

ML-research



Impact. Вместо резюме

15

Пробивших потолок разработчиков

- 4
Прокачанных техлидов

- 1
Замотивированный Антиспам



Ask Me Nicely

d.merkushov@corp.mail.ru
https://t.me/dmerkushov



